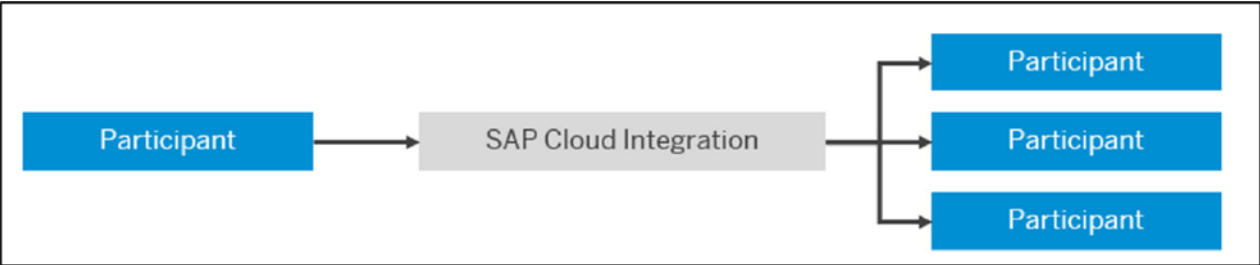


EXHIBIT 9


Exhibit 9: U.S. Patent No. 6,889,244

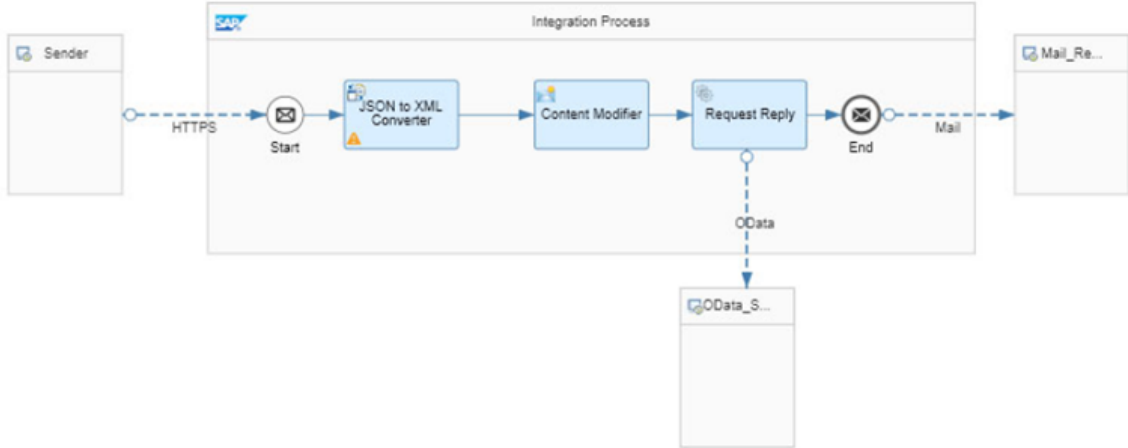
Claim 1	Identification
<p>1[pre]. A method of transmitting messages between a first node and a second node, wherein the first and second nodes are each coupled to a fault tolerant storage system (FTSS), the method comprising:</p>	<p>To the extent the preamble is limiting, SAP performs a method of transmitting messages between a first node and a second node, wherein the first and second nodes are each coupled to a fault tolerant storage system (FTSS). For example, <i>see</i>:</p> <div data-bbox="615 440 1268 505" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>What Is SAP Cloud Integration?</p> </div> <div data-bbox="615 521 1887 716" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Support end-to-end process integration through the exchange of messages.</p> <p>SAP Cloud Integration helps you to connect cloud and on-premise applications with other SAP and non-SAP cloud and on-premise applications. This service can process messages in real-time scenarios spanning different companies, organizations, or departments within one organization.</p> </div> <div data-bbox="615 727 1887 849" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Set up secure and reliable communication</p> <p>Use our core integration and security capabilities for the safe and reliable processing of messages. Configure how messages are exchanged within an integration scenario so that the data involved is protected according to the newest security standards.</p> </div> <div data-bbox="615 860 1887 982" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Connect to multiple endpoints</p> <p>Integrate various applications and data sources from SAP and non-SAP, on premise, as well as the cloud. SAP Cloud Integration comes with a set of prebuilt adapters.</p> </div> <p>Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/what-is-sap-cloud-integration.</p> <div data-bbox="615 1089 846 1138" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Introduction</p> </div> <div data-bbox="615 1154 1887 1235" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>You can connect various kinds of remote systems to the cloud-based integration platform using protocols such as HTTP/S, SSH and SMTP/S. Each communication protocol comes with certain options to protect the message exchange (security options).</p> </div> <p>Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/introduction.</p>


Claim 1	Identification								
	<div data-bbox="611 266 1864 529">  <pre> graph LR P1[Participant] --> SCIS[SAP Cloud Integration] SCIS --> P2[Participant] SCIS --> P3[Participant] SCIS --> P4[Participant] </pre> </div> <p data-bbox="611 532 1864 570">Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/integration-capabilities</p> <div data-bbox="611 602 1803 751"> <p>Cloud Integration supports various integration patterns. Certain integration patterns require a storage location to read and write the content of a message during the execution of a scenario. When the message content is needed for later processing steps or if the message content is evaluated in a later phase, storage steps are required. Storing data is also required in asynchronous message patterns. In this section, we discuss how to use the following storage options (in the context of different use cases):</p> </div> <p data-bbox="611 755 1864 824">Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/data-store-variables-and-jms-queues-when-to-use-which-option.</p> <div data-bbox="611 862 1803 1304"> <table border="1"> <thead> <tr> <th data-bbox="625 889 911 927">Feature</th><th data-bbox="911 889 1789 927">Description</th></tr> </thead> <tbody> <tr> <td data-bbox="625 927 911 987">Persist Message</td><td data-bbox="911 927 1789 987">Stores a message payload so that you can access the stored message and analyze it at a later point in time.</td></tr> <tr> <td data-bbox="625 987 911 1247">Data Store Operations</td><td data-bbox="911 987 1789 1247"> <p>Stores messages temporarily for later processing.</p> <p>The following operations are supported:</p> <ul style="list-style-type: none"> ▪ SELECT ▪ GET ▪ WRITE ▪ DELETE </td></tr> <tr> <td data-bbox="625 1247 911 1300">Write Variables</td><td data-bbox="911 1247 1789 1300">Specifies values for variables required during message processing.</td></tr> </tbody> </table> </div> <p data-bbox="611 1307 1864 1344">Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/integration-capabilities</p>	Feature	Description	Persist Message	Stores a message payload so that you can access the stored message and analyze it at a later point in time.	Data Store Operations	<p>Stores messages temporarily for later processing.</p> <p>The following operations are supported:</p> <ul style="list-style-type: none"> ▪ SELECT ▪ GET ▪ WRITE ▪ DELETE 	Write Variables	Specifies values for variables required during message processing.
Feature	Description								
Persist Message	Stores a message payload so that you can access the stored message and analyze it at a later point in time.								
Data Store Operations	<p>Stores messages temporarily for later processing.</p> <p>The following operations are supported:</p> <ul style="list-style-type: none"> ▪ SELECT ▪ GET ▪ WRITE ▪ DELETE 								
Write Variables	Specifies values for variables required during message processing.								

Claim 1	Identification
	<div data-bbox="611 266 1698 651"> <h3 data-bbox="636 285 829 321">Resilience</h3> <p data-bbox="636 358 1690 639">One of the most crucial qualities of an enterprise-grade integration flow is its resilience. The traditional (deterministic) approach assumes a software system free of malfunctions. But resilient software design doesn't try to avoid failures. Instead, it assumes that failures are going to occur and can't be prevented or predicted. The resilience approach tries to minimize the time that elapses between the occurrence of a failure and its correction. An integration flow can involve complex integration logic including enriching the message content with data from external components, storing data in a database persistence, and much more. Consequently, make sure that you apply principles for resilient software development when developing an integration flow.</p> </div> <p data-bbox="611 656 1829 721">Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/guidelines-to-design-enterprise-grade-integration-flows</p> <div data-bbox="611 760 1801 1317"> <h3 data-bbox="630 776 1104 818">Apply the Retry Pattern</h3> <p data-bbox="630 855 1577 876">Apply the retry pattern for handling anticipated, temporary failures when connecting to an external component.</p> <p data-bbox="630 911 1793 959">If an error occurs during the processing of an asynchronous scenario, the sender component usually gets an error and has to handle the reprocessing of the message at a later point in time. The sender system needs to contain a retry mechanism.</p> <p data-bbox="630 993 1766 1042">A better method is to trigger retries directly from SAP Cloud Integration. The sender, in such a case, simply delivers the message and doesn't need to worry about retries if error occurs. Retry is handled by the SAP Cloud Integration component.</p> <p data-bbox="630 1089 1793 1198">SAP Cloud Integration offers storage to persist data in transit during message processing, namely the message queue storage used by the JMS adapter. Use this storage to persist the message at the beginning of the processing sequence. That way, processing is executed faster for the sender, who immediately receives a response with HTTP code 202 (Accepted), and the subsequent processing steps are executed asynchronously.</p> <p data-bbox="630 1227 1793 1308">With this storage option, a retry mechanism is also in place that works as follows: If message processing fails due to a temporary error when calling an external component, the storage can be used to persist the failed messages. The JMS adapter polls the storage regularly for content, and triggers the reprocessing of the respective messages.</p> </div> <p data-bbox="611 1325 1829 1390">Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/guidelines-to-design-enterprise-grade-integration-flows.</p>

Claim 1	Identification
	<div>Using Data Storage Features When Designing Integration Flows</div> <div>SAP Cloud Integration comes with various options for storing data during the execution of an integration scenario. There are also various options for consuming such data during message processing or after message processing has finished.</div> <div>The following figure illustrates the available options.</div> <div><p>The diagram illustrates the data storage options available in SAP Cloud Integration. It shows a Tenant environment with the following components:</p><ul style="list-style-type: none">Integration Flow Processing: Message Header, Message Body, Exchange Property.Monitoring Data and Message Store: Message store entry, Message processing log.Runtime Data and Variables: Variable, Data store entry, Partner Directory content.JMS Queues: Message.<p>These components are connected to various external systems:</p><ul style="list-style-type: none">Remote Archiving System: Write data: Monitoring application; Read data: Monitoring application.Remote Database: Write data: Data store Write, Write variables, Aggregator, Script, XI adapter, Partner Directory API; Read data: Content modifier, Data store Select/Get/Delete, Data store sender adapter, OData API, Script.Remote Kafka Server: Write data: JMS/XI /AS2 receiver adapter; Read data: JMS/XI/AS2 sender adapter, Monitoring application, OData API.Remote Message Broker: Write data: Monitoring application; Read data: JMS/XI/AS2 sender adapter, Monitoring application, OData API.Remote Logging Solution (e.g. Splunk): Write data: API; Read data: API.</div>

Claim 1	Identification
	<p>The options also differ according to the phase in which components can consume the stored data:</p> <ul style="list-style-type: none"> • Stored data consumed during message processing (at runtime): For example, if you want to store a variable that, and another step consumes the data during consecutive processing of the integration flow. • Stored data consumed after message processing has finished: For example, if you want to store data and make it available for auditing purposes after message processing has finished. <p>Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/using-data-storage-features-when-designing-integration-flows.</p>
<p>1[a]. transmitting a message from the first node to a communication agent in the FTSS;</p>	<p>SAP transmits a message from the first node to a communication agent in the FTSS. For example, <i>see</i>:</p> <div data-bbox="611 716 1885 1062"> <p>SAP Cloud Integration supports various integration patterns, or ways how applications can be integrated with each other.</p> <p>The following figure illustrates, as one example, the routing pattern, that allows you to forward a message from one participant to multiple receivers.</p>  <pre> graph LR P1[Participant] --> S[SAP Cloud Integration] S --> P2[Participant] S --> P3[Participant] S --> P4[Participant] </pre> </div> <p>Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/integration-capabilities.</p>

Claim 1	Identification
	<div data-bbox="611 266 1759 1133"> <h2 data-bbox="625 277 1612 396">Sender-Initiated Scenario (with HTTPS Sender Adapter)</h2> <p data-bbox="625 440 1696 467">Create a simple integration scenario that is initiated by a sender (using the HTTPS sender adapter).</p> <p data-bbox="625 508 1738 573">With the following steps, you can easily modify and extend the previously built integration flow with the email receiver (<i>Timer-Initiated Scenario with a Mail Receiver</i>).</p> <p data-bbox="625 613 1528 641">The figure shows the integration flow model that you get as a result of this exercise.</p>  <pre> graph LR Sender[Sender] -- HTTPS --> Start((Start)) Start --> Converter[JSON to XML Converter] Converter --> Modifier[Content Modifier] Modifier --> RequestReply[Request Reply] RequestReply --> End((End)) End -- Mail --> MailReceiver[Mail_Receiver] RequestReply -. OData .-> ODataSender[OData_Sender] </pre> </div> <p data-bbox="604 1138 1885 1208">Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/sender-initiated-scenario-with-https-sender-adapter.</p> <div data-bbox="611 1247 1759 1398"> <h2 data-bbox="632 1258 1367 1300">Assign Sender and Receiver Components</h2> <p data-bbox="632 1328 1745 1386">You use the Sender and Receiver elements to model remote systems that are connected to your integration flow (either as sender or receiver of messages).</p> </div>

Claim 1	Identification
	<p>Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/assign-sender-and-receiver-components.</p> <div data-bbox="611 337 1879 1094" style="border: 1px solid black; padding: 10px;"> <h3 style="text-align: center;">Connecting a Customer System to Cloud Integration</h3> <p>You can set up the technical connection between a tenant and different kinds of remote systems (in many cases located in the customer landscape).</p> <p>Throughout this documentation we assume the following basic setup of technical components and communication paths: A remote system (which is not specified) is being connected to one of the tenants that are assigned to the customer. The remote system can act either as a sender or a receiver of messages. The setup and the detailed configuration procedure differ according to the communication direction that is being set up: whether a remote system is supposed to send a message to the integration platform or the other way round.</p> <p>Throughout this documentation, the terms <i>inbound</i> and <i>outbound</i> reflect the perspective of the integration platform.</p> <ul style="list-style-type: none"> • Inbound refers to message processing from a remote system (in many cases, located in the customer landscape) to Cloud Integration. Here, the integration platform is the server. • Outbound refers to message processing from the integration platform to a remote system (where the integration platform is the client).  <pre> graph LR RS1[Remote System] -- "Inbound Communication" --> CI[Cloud Integration] CI -- "Outbound Communication" --> RS2[Remote System] </pre> </div> <p>Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/connecting-customer-system-to-cloud-integration.</p>

Claim 1	Identification
	<div data-bbox="611 266 1801 834"><h3 data-bbox="636 282 1318 315">Kind of Systems to Connect to Cloud Integration</h3><p data-bbox="636 347 1745 404">To give you an idea of which kinds of remote systems can be connected to the integration platform, here are some typical examples (this is not a complete list):</p><ul data-bbox="678 440 1430 615" style="list-style-type: none"><li data-bbox="678 440 1373 464">• On-premise systems, for example, SAP systems based on SAP NetWeaver<li data-bbox="678 488 821 513">• SFTP servers<li data-bbox="678 537 1430 561">• Cloud applications, for example, SAP SuccessFactors or SAP Cloud for Customer<li data-bbox="678 586 1188 610">• Other systems such as e-mail servers or SOAP clients<p data-bbox="636 651 1772 708">Depending on the kind of system to connect, a certain communication protocol is to be considered, as will be explained in the next section.</p><p data-bbox="636 740 1793 826">To support dedicated kinds of systems (through dedicated communication protocols), the integration platform provides certain adapters. An adapter allows you to configure the details of the technical communication channel between the remote system and the integration platform.</p></div> <p data-bbox="611 842 1713 875">Source: <a data-bbox="716 842 1713 875" href="https://help.sap.com/docs/cloud-integration/sap-cloud-integration/introduction">https://help.sap.com/docs/cloud-integration/sap-cloud-integration/introduction</p>

Claim 1	Identification
	<p>To give you an idea of what the technical landscape behind a real life integration scenario looks like, here's an example for the SAP Cloud for Customer (C4C)-to-SAP ERP integration scenario. In this scenario, SAP's own cloud solution SAP Cloud for Customer (C4C) is connected with an on-premise SAP Enterprise Resource Planning (ERP) system through Cloud Integration.</p> <p>The following figure shows a typical setup of components:</p> <p>The left side of the figure covers the communication of Cloud Integration with the on-premise system in the customer landscape.</p> <p>The setup contains components that all are connected by HTTPS communication. Typical adapters are the IDoc adapter for the connection between the on-premise system and Cloud Integration, and the SOAP adapter for the connection between SAP Cloud for Customer and Cloud Integration (within the SAP Cloud).</p> <p>The lower path shows the connection from Cloud Integration to the on-premise system, which is located in the customer landscape. This is the outbound communication from the perspective of the integration platform, but is an inbound connection from the perspective of the customer landscape. Therefore, to protect the components in the customer landscape from remote calls from the Internet, a load balancer component is required – which is either a Web Dispatcher component or the SAP Cloud Connector.</p> <p>The upper path shows the connection from the on-premise system to Cloud Integration. From the <i>perspective</i> of Cloud Integration, this is an inbound connection and, therefore, again a load balancer is required to protect the tenant that actually processes the message against remote calls. This is the BIG-IP load balancer, which is involved in all HTTPS inbound requests by default, and isn't shown in the figure for the sake of simplicity. Also, this component is preconfigured by SAP and doesn't require any further configuration for such a scenario.</p>

Claim 1	Identification										
	<p>Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/connecting-to-on-premise-landscape-example-setup.</p> <div data-bbox="611 337 1801 506" style="border: 1px solid black; padding: 10px;"> <h3 style="margin: 0;">IDoc Adapter</h3> <p style="margin: 0;">The IDoc adapter enables SAP Cloud Integration to exchange Intermediate Document (IDoc) messages with systems that support communication via SOAP Web services.</p> </div> <p>Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/idoc-adapter.</p>										
<p>1[b]. storing the message in a data structure in highly reliable fault tolerant storage media of the FTSS;</p>	<p>SAP stores the message in a data structure in highly reliable fault tolerant storage media of the FTSS. For example, <i>see</i>:</p> <div data-bbox="611 691 1801 841" style="border: 1px solid black; padding: 10px;"> <p>Cloud Integration supports various integration patterns. Certain integration patterns require a storage location to read and write the content of a message during the execution of a scenario. When the message content is needed for later processing steps or if the message content is evaluated in a later phase, storage steps are required. Storing data is also required in asynchronous message patterns. In this section, we discuss how to use the following storage options (in the context of different use cases):</p> </div> <p>Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/data-store-variables-and-jms-queues-when-to-use-which-option.</p> <div data-bbox="611 951 1845 1399" style="border: 1px solid black; padding: 10px;"> <table> <tr> <th colspan="2">Storing Data During Processing</th></tr> <tr> <th>Feature</th><th>Description</th></tr> <tr> <td>Persist Message</td><td>Stores a message payload so that you can access the stored message and analyze it at a later point in time.</td></tr> <tr> <td>Data Store Operations</td><td> <p>Stores messages temporarily for later processing.</p> <p>The following operations are supported:</p> <ul style="list-style-type: none"> ▪ SELECT ▪ GET ▪ WRITE ▪ DELETE </td></tr> <tr> <td>Write Variables</td><td>Specifies values for variables required during message processing.</td></tr> </table> </div>	Storing Data During Processing		Feature	Description	Persist Message	Stores a message payload so that you can access the stored message and analyze it at a later point in time.	Data Store Operations	<p>Stores messages temporarily for later processing.</p> <p>The following operations are supported:</p> <ul style="list-style-type: none"> ▪ SELECT ▪ GET ▪ WRITE ▪ DELETE 	Write Variables	Specifies values for variables required during message processing.
Storing Data During Processing											
Feature	Description										
Persist Message	Stores a message payload so that you can access the stored message and analyze it at a later point in time.										
Data Store Operations	<p>Stores messages temporarily for later processing.</p> <p>The following operations are supported:</p> <ul style="list-style-type: none"> ▪ SELECT ▪ GET ▪ WRITE ▪ DELETE 										
Write Variables	Specifies values for variables required during message processing.										

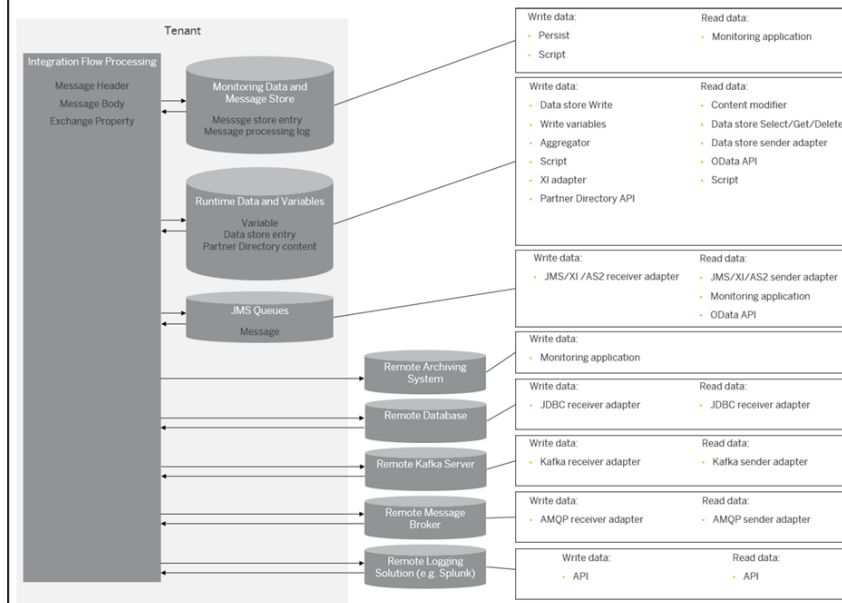
Claim 1	Identification
	<p>Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/integration-capabilities</p> <div data-bbox="611 305 1732 699" style="border: 1px solid black; padding: 10px;"> <p>Resilience</p> <p>One of the most crucial qualities of an enterprise-grade integration flow is its resilience. The traditional (deterministic) approach assumes a software system free of malfunctions. But resilient software design doesn't try to avoid failures. Instead, it assumes that failures are going to occur and can't be prevented or predicted. The resilience approach tries to minimize the time that elapses between the occurrence of a failure and its correction. An integration flow can involve complex integration logic including enriching the message content with data from external components, storing data in a database persistence, and much more. Consequently, make sure that you apply principles for resilient software development when developing an integration flow.</p> </div> <p>Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/guidelines-to-design-enterprise-grade-integration-flows</p> <div data-bbox="611 808 1661 1073" style="border: 1px solid black; padding: 10px;"> <p>Apply the Retry Pattern</p> <p>Apply the retry pattern for handling anticipated, temporary failures when connecting to an external component.</p> <p>If an error occurs during the processing of an asynchronous scenario, the sender component usually gets an error and has to handle the reprocessing of the message at a later point in time. The sender system needs to contain a retry mechanism.</p> <p>A better method is to trigger retries directly from SAP Cloud Integration. The sender, in such a case, simply delivers the message and doesn't need to worry about retries if error occurs. Retry is handled by the SAP Cloud Integration component.</p> </div> <div data-bbox="611 1084 1661 1300" style="border: 1px solid black; padding: 10px;"> <p>SAP Cloud Integration offers storage to persist data in transit during message processing, namely the message queue storage used by the JMS adapter. Use this storage to persist the message at the beginning of the processing sequence. That way, processing is executed faster for the sender, who immediately receives a response with HTTP code 202 (Accepted), and the subsequent processing steps are executed asynchronously.</p> <p>With this storage option, a retry mechanism is also in place that works as follows: If message processing fails due to a temporary error when calling an external component, the storage can be used to persist the failed messages. The JMS adapter polls the storage regularly for content, and triggers the reprocessing of the respective messages.</p> </div> <p>Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/guidelines-to-design-enterprise-grade-integration-flows.</p>

Claim 1**Identification**

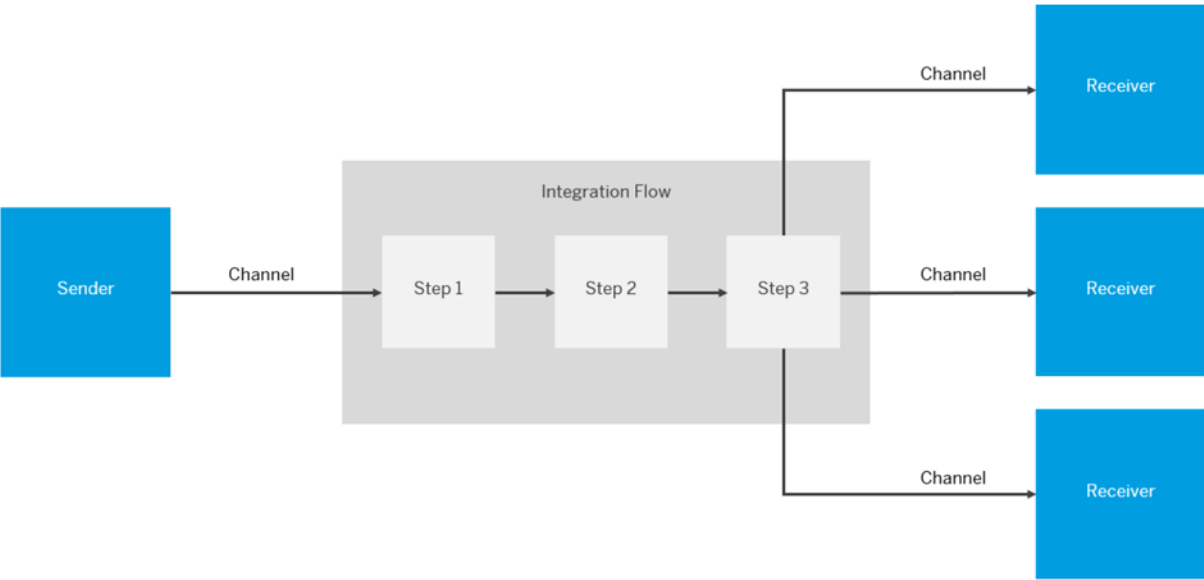
Using Data Storage Features When Designing Integration Flows

SAP Cloud Integration comes with various options for storing data during the execution of an integration scenario. There are also various options for consuming such data during message processing or after message processing has finished.

The following figure illustrates the available options.



Claim 1	Identification
	<p>The options also differ according to the phase in which components can consume the stored data:</p> <ul style="list-style-type: none"> • Stored data consumed during message processing (at runtime): For example, if you want to store a variable that, and another step consumes the data during consecutive processing of the integration flow. • Stored data consumed after message processing has finished: For example, if you want to store data and make it available for auditing purposes after message processing has finished. <p>Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/using-data-storage-features-when-designing-integration-flows.</p>
<p>1[c]. processing the message at the FTSS in accordance with a messaging paradigm; and</p>	<p>SAP processes the message at the FTSS in accordance with a messaging paradigm. For example, <i>see</i>:</p> <div data-bbox="611 643 1864 1127" style="border: 1px solid black; padding: 10px;"> <h3 style="text-align: center;">What Is Integration Content</h3> <p>Operating business processes using SAP Cloud Integration implies the exchange of data (messages) between the participants. How messages are exchanged is specified by integration content that is designed based on the requirements of the business process.</p> <p>As one key part of integration content, integration flows describe how a message sent from one participant is processed by SAP Cloud Integration.</p> <p>In other words, using integration flows, specific integration pattern like mapping or routing can be specified.</p> <p>For example, a set of integration flows specifies that a message sent from participant A is forwarded by SAP Cloud Integration to three different receivers B, C, and D, dependent on the business content contained in message. Integration flows also specify mappings of the data structure between sender and receiver or the endpoints of sender and receiver participants.</p> </div> <p>Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/what-is-integration-content.</p>


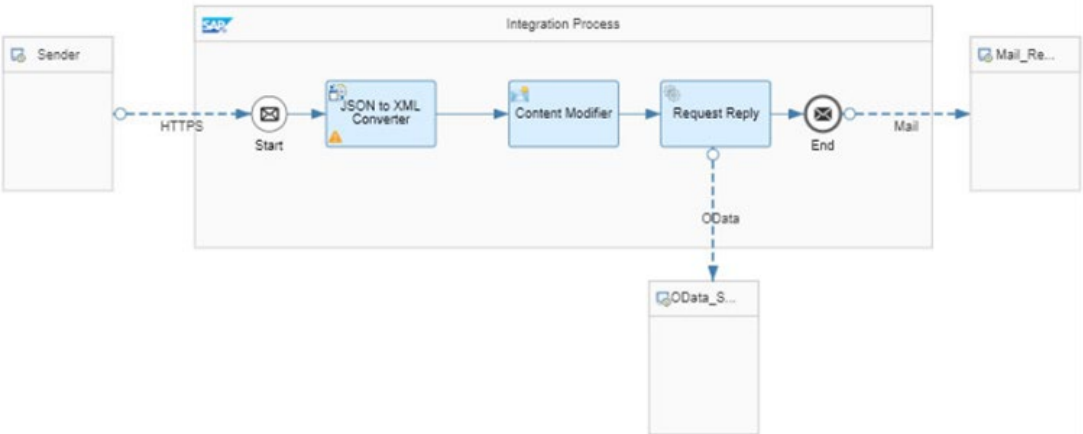
Claim 1	Identification
	<p data-bbox="646 280 898 313">Integration Flows</p> <p data-bbox="646 345 1411 370">An integration flow allows you to specify how a message is processed on a tenant.</p> <p data-bbox="646 407 1535 431">The following figure provides a simplified and generalized representation of an integration flow.</p>  <p>The diagram illustrates an integration flow. On the left, a blue square labeled 'Sender' has an arrow labeled 'Channel' pointing to a gray rectangular box labeled 'Integration Flow'. Inside this box, three light gray squares are arranged horizontally, labeled 'Step 1', 'Step 2', and 'Step 3', connected by arrows. From 'Step 3', an arrow labeled 'Channel' points to a blue square labeled 'Receiver'. Additionally, two more arrows labeled 'Channel' branch off from the right side of the 'Integration Flow' box, each pointing to a separate blue square labeled 'Receiver'.</p> <p data-bbox="611 1032 1881 1101">Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/elements-of-cloud-based-integration-scenario.</p>


Claim 1	Identification
	<div data-bbox="611 305 1866 475"> <h3>Guidelines to Implement Specific Integration Patterns</h3> <p>Cloud Integration supports the implementation of enterprise integration patterns that are also referred to as <i>integration patterns</i> or <i>messaging patterns</i>.</p> </div> <div data-bbox="611 492 982 925"> <ul style="list-style-type: none"> Aggregator Composed Message Processor Content-Based Routing Content Enricher Content Filter Message Filter Recipient List Resequencer Scatter-Gather Splitter Quality of Service Exactly Once </div> <p data-bbox="611 933 1881 1003">Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/guidelines-to-implement-specific-integration-patterns.</p> <div data-bbox="611 1076 1866 1284"> <h3>Aggregator</h3> <p>You want to combine related individual messages so that they can be processed in bulk. Using an Aggregator pattern, you can collect and store individual messages until a complete set of related messages has been received. The aggregated message is then sent to the actual receiver.</p> </div> <p data-bbox="611 1292 1701 1323">Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/aggregator.</p>

Claim 1	Identification																																	
	<div><p>Table 3: Sample Integration Patterns – Integration and Application Layer</p><table><tr><th>Sample integration pattern</th><th>Integration layer</th><th>Application layer</th></tr><tr><th>Integration technology</th><th>Cloud Integration capability of SAP Integration Suite or SAP Process Orchestration</th><th>SAP Application Interface Framework</th></tr><tr><td>Structure mapping</td><td>●</td><td>●</td></tr><tr><td>Routing</td><td>●</td><td>●</td></tr><tr><td>Splitter</td><td>●</td><td>●</td></tr><tr><td>Aggregator</td><td>●</td><td>●</td></tr><tr><td>Protocol adaptation</td><td>●</td><td>●</td></tr><tr><td>Value mapping – static code list</td><td>●</td><td>●</td></tr><tr><td>Value mapping – dynamic code list</td><td>●</td><td>●</td></tr><tr><td>Data validation</td><td>●</td><td>●</td></tr><tr><td>Forward error correction (business user)</td><td>●</td><td>●</td></tr></table><p>● Good fit ● Partial fit ● Not a fit</p><div><p>It is recommended that the following integration patterns be implemented in the integration layer either in the Cloud Integration capability of the integration suite or SAP Process Orchestration, as reflected in Table 3.</p><ul style="list-style-type: none">• The structure mapping pattern maps message fields between two interfaces. Structure mappings can mediate between different structures of the same format (for example, two different XML structures) or even change the format (for example, from JSON to XML).• The routing pattern is used to determine for a given message a set of receivers based on static settings (technical routing) or message payload data (content-based routing). Logical receiver destinations define a message receiver within the sending application. The determination of logical receiver destinations within an application is usually tightly coupled with the application data and process model and, hence, is part of the application layer. An example is the configuration of the distribution model in the ABAP⁴ programming language.• The splitter pattern creates multiple messages out of a received message by fragmenting the payload by a splitter rule. By then applying the routing pattern, split message parts might be routed to different receivers.• The aggregator pattern collects multiple inbound messages until a modeled threshold is reached. A single payload is then constructed out of all aggregated payloads by applying an aggregation rule. Finally, this single message is forwarded to one or more receivers.• The protocol adaptation pattern handles the mediation between two different protocols used by the sending and receiving applications (for example, Java Message Service, REST, or SOAP⁴).<p>⁴ List of abbreviations can be found at the end of the document.</p><p><< 29 / 99 >></p></div></div>	Sample integration pattern	Integration layer	Application layer	Integration technology	Cloud Integration capability of SAP Integration Suite or SAP Process Orchestration	SAP Application Interface Framework	Structure mapping	●	●	Routing	●	●	Splitter	●	●	Aggregator	●	●	Protocol adaptation	●	●	Value mapping – static code list	●	●	Value mapping – dynamic code list	●	●	Data validation	●	●	Forward error correction (business user)	●	●
Sample integration pattern	Integration layer	Application layer																																
Integration technology	Cloud Integration capability of SAP Integration Suite or SAP Process Orchestration	SAP Application Interface Framework																																
Structure mapping	●	●																																
Routing	●	●																																
Splitter	●	●																																
Aggregator	●	●																																
Protocol adaptation	●	●																																
Value mapping – static code list	●	●																																
Value mapping – dynamic code list	●	●																																
Data validation	●	●																																
Forward error correction (business user)	●	●																																

Source: Integration Architecture Guide for Cloud and Hybrid Landscapes Based on SAP Integration Solution Advisory Methodology at 29 (available at <https://www.sap.com/documents/2020/11/400ae14b-bf7d-0010-87a3-c30de2ffd8ff.html>)

Claim 1	Identification
	<div data-bbox="611 269 1793 1084" style="border: 1px solid black; padding: 10px;"> <p>For further integration patterns related to the integration layer, refer to the Enterprise Integration Patterns Web site. How to implement commonly used enterprise integration patterns using the Cloud Integration capability of SAP Integration Suite is described in a rich set of integration flow guidelines and patterns as part of the user documentation and published on SAP API Business Hub.</p> <p>The following patterns can be implemented either in the integration layer or the application layer:</p> <ul style="list-style-type: none"> • The value mapping pattern transforms field values that are exchanged between two applications. Since this pattern is tightly coupled with business application data, business users must be able to maintain value mappings in a dynamic way. The value mapping pattern is typically covered by SAP Application Interface Framework, as business users may detect application errors, such as incomplete value mappings, due to unforeseen values coming up. They then must extend these mappings in an agile way close to the application. As opposed to dynamic value mappings, static code-list-value mappings (for example, country code mappings) can be best implemented in the integration layer, as they can be reused across multiple integration scenarios between various applications. • The data validation pattern checks a received message with respect to schema correctness or data correctness. This pattern can be applied in the integration layer as well as in the application layer. SAP Application Interface Framework particularly supports this pattern, as it provides full access to application data and functions. • The forward error correction pattern handles an application error during message processing by forwarding it to the receiver. One key capability of SAP Application Interface Framework (in the application layer) is this exact type of error correction and resolution by business users. </div> <p>Source: Integration Architecture Guide for Cloud and Hybrid Landscapes Based on SAP Integration Solution Advisory Methodology at 30 (available at https://www.sap.com/documents/2020/11/400ae14b-bf7d-0010-87a3-c30de2ffd8ff.html)</p>
1[d]. transmitting the message from the FTSS to the second node.	SAP transmits the message from the FTSS to the second node. For example, <i>see</i> :

Claim 1	Identification
	<p data-bbox="636 285 1661 310">SAP Cloud Integration supports various integration patterns, or ways how applications can be integrated with each other.</p> <p data-bbox="636 339 1808 391">The following figure illustrates, as one example, the routing pattern, that allows you to forward a message from one participant to multiple receivers.</p>  <p data-bbox="611 610 1850 639">Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/integration-capabilities</p> <p data-bbox="636 691 1371 732">Assign Sender and Receiver Components</p> <p data-bbox="636 764 1759 816">You use the Sender and Receiver elements to model remote systems that are connected to your integration flow (either as sender or receiver of messages).</p> <p data-bbox="611 849 1797 914">Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/assign-sender-and-receiver-components.</p> 

Claim 1	Identification
	<p>Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/sender-initiated-scenario-with-https-sender-adapter.</p> <div data-bbox="611 337 1642 954"> <h3>Connecting a Customer System to Cloud Integration</h3> <p>You can set up the technical connection between a tenant and different kinds of remote systems (in many cases located in the customer landscape).</p> <p>Throughout this documentation we assume the following basic setup of technical components and communication paths: A remote system (which is not specified) is being connected to one of the tenants that are assigned to the customer. The remote system can act either as a sender or a receiver of messages. The setup and the detailed configuration procedure differ according to the communication direction that is being set up: whether a remote system is supposed to send a message to the integration platform or the other way round.</p> <p>Throughout this documentation, the terms <i>inbound</i> and <i>outbound</i> reflect the perspective of the integration platform.</p> <ul style="list-style-type: none"> • Inbound refers to message processing from a remote system (in many cases, located in the customer landscape) to Cloud Integration. Here, the integration platform is the server. • Outbound refers to message processing from the integration platform to a remote system (where the integration platform is the client).  <pre> graph LR RS1[Remote System] -- "Inbound Communication" --> CI[Cloud Integration] CI -- "Outbound Communication" --> RS2[Remote System] </pre> </div> <p>Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/connecting-customer-system-to-cloud-integration.</p> <div data-bbox="611 1068 1736 1235"> <p>For each protocol, different authentication options are supported - ways how the connected systems prove their trustworthiness against each other during connection setup. Connection setup is performed differently, depending on whether inbound communication (when a remote system as a sender calls Cloud Integration) or outbound communication (when Cloud Integration calls a remote system which, in turn, is then considered as the receiver) is configured. The detailed procedure also depends on the chosen protocol and authentication option.</p> </div> <p>Source: https://help.sap.com/docs/cloud-integration/sap-cloud-integration/introduction.</p>